BACnet/OPC-Server

User manual

AMANN GmbH Digitale Regelungstechnik
Manager: Oliver Riesenberger
Hauptstrasse 42
82008 Unterhaching
Telefon: +49 (0)89 - 613 937-0
Telefax: +49 (0)89-613 937-99
www.amann-net.de

rev: 2005.1 V1.55.00E 3.2.2005

1 INTRODUCTION 3

2 INSTALLATION 4

2.1 System requirements 4

2.2 Installation 4

3 CONFIGURATION FILE. 6

3.1 Overview of configuration file structure. 6

3.2 Example configuration file 7

3.3 Description of parameters 8
3.3.1 General Server-parameters 8
3.3.2 Device-Parameter 12
3.3.3 Object-Parameter 12

4 SERVER START UP AND SHUTDOWN 15

5 RUNTIME 16

5.1 Log files 16

5.2 Online status 17

5.3 Polls and COV 17

5.4 Demo mode 19

6 APPENDIX 20

6.1 Default values 20

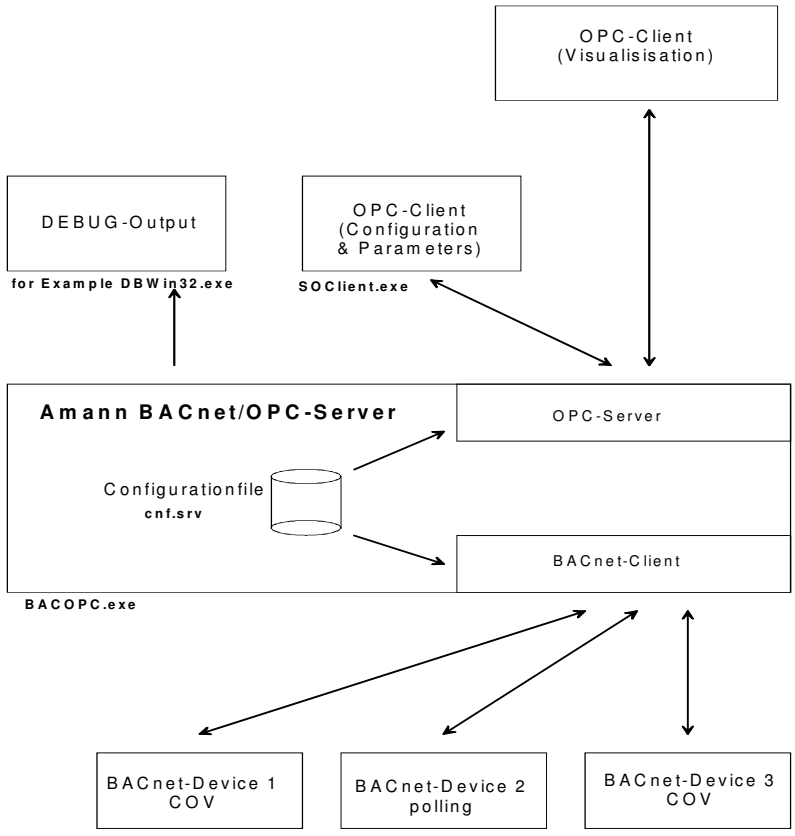
6.2 Files 21

6.3 BACnet-specific Error-codes 22

7 ERROR HANDLING 23

1 Introduction

The Amann BACnet/OPC Server enables communication between BACnet devices and OPC Clients. The Server is automatically started if required by an OPC client. At start up a configuration file is read which defines the BACnet devices, objects and parameters with which the OPC Server will communicate. Configuration of the Server can be done simply at a global level using defaults, at a device level or timing can be defined at an object level. The Amann BACnet/OPC Server supports COV (change of value) and polling. Up to 50,000 OPC objects can be handled. A debugging and monitoring feature of the Server enables status and error messages to be broadcast in real time to a Windows-debug-interface and also written into log files
During runtime the BACnet/OPC Server can be tested with the help of an OPC Client (SOClient.exe) which is installed during the installation of the BACnet/OPC Server.
In the field of the building automation the numbers of data points increases continuously. Therefore a great effort has been made to develop a high performance application which needs minimal resources.



Einbindung des Amann BACnet/OPC-Servers

2 Installation

2.1 System requirements

Recommended PC requirements for the Amann BACnet/OPC Server

- **Windows NT with service Pack 5 (not , Windows 2000 or Windows XP**
  Note: Running BACnet over IP currently requires Windows XP
- **100 Mbytes of RAM is required for approximately 20,000 data points, the memory requirement is dependent on the number data points in use.**

Attention: You must run BacOpc (or BACBrowse) at least once as administrator after booting your computer!

2.2 Installation

Run **`Setup.exe`** from the distribution CD.

To check that the installation was successful and the Server was registered properly

- Run the “DCOMCNFG.exe” application (Start->Run->”DCOMCNFG”).
- Verify that the entry “Ammans OPCServer” is available under “DCOM Config” as shown in figure 2.2.1 (XP) or figure 2.2.2 (NT).

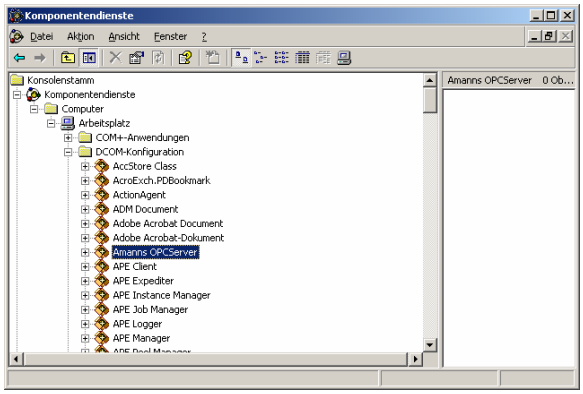


Figure 2.2.1. Windows XP DCOM configuration.

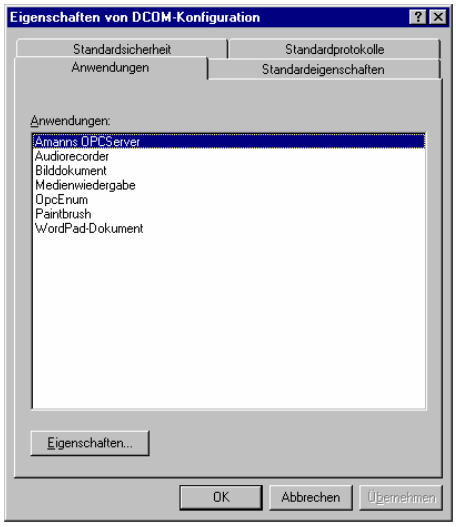
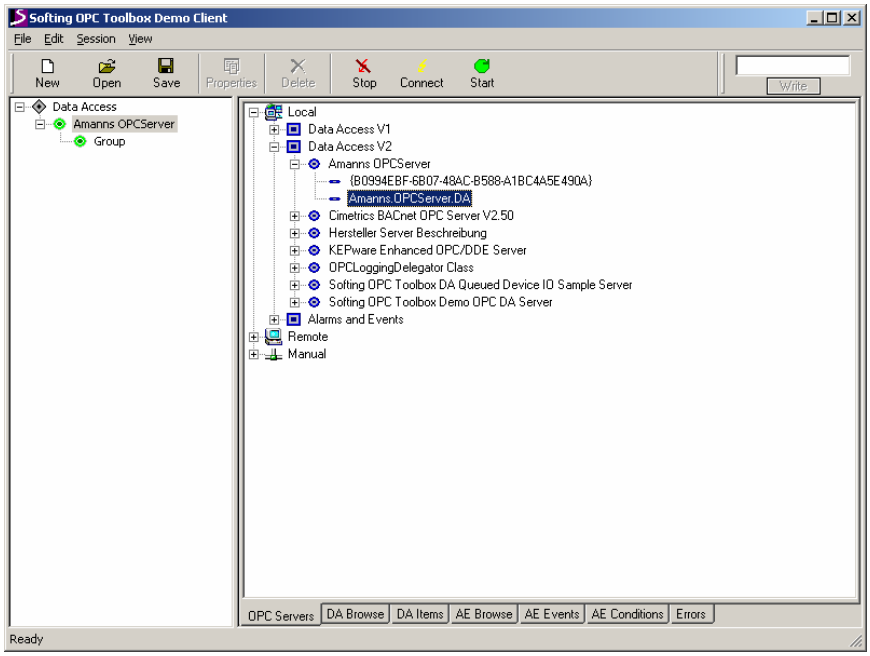


Figure 2.2.2. Windows NT DCOM configuration (under applications).

During the Amann BACnet/OPC Server installation an OPC client “SOClient.exe” is also installed (figure 2.2.3). This can be used to test the Amann BACnet/OPC Server by reading and writing to datapoints in the Server OPC Namespace



3 Configuration file.

The configuration of the Amann BACnet/OPC Server is done by the “cnf.srv” configuration file which is read only when the Server starts. The location of the “cnf.srv” configuration file is always in the same directory as the BACOPC.exe. The configuration file contains all BACnet devices, data points and special values for controlling the behaviour of the Server. During the installation process an example “cnf.srv” configuration file is provided that has one BACnet Device with BacnetID 3400 and COV behaviour.

3.1 Overview of configuration file structure.

The configuration file is separated into the following sections or OPC namespace objects (each of these strting by the keyword “begin”):

Apduprops (general Server settings)	begin APDU
Device-parameter 1	begin device
Objekt-Parameter 1	begin object
Objekt-Parameter 2	begin object
…	…
Objekt-Parameter n	begin object
Device-parameter 2	begin device
Objekt-Parameter 1	begin object
Objekt-Parameter 2	begin object
…	…
Objekt-Parameter n	begin object
…	
Device-Parameter n	

A comment can be insterted by preceeding it with a double slash (“//”). All typing until the line end is ignored.

Nomenclature of *Parameter* :

Parametername [: R | W [= Wert]]

Parametername is one of the summarized keywords below, such as : cov_process_id or present-value. It is optionally followed by “: R” or “: “ indicating the property being read only (default) or writeable.

Note: This flag defines the writeability of the OPC reference object. The OPC-server will forward a write request to the BACnet server. Whether or not the BACnet property is actual writeable, depends on the BACnet server.

Furthermore an equal sign may follow indicating an initial value. This value will be passed to OPC clients during startup until the actual value gets retrieved from BACnet (which might take a longer in large installations) .

The configuration file largely doesn't need a special formatting. Spaces and even line breaks may get freely used in most places (exception: Strings and comments).

3.2 Example configuration file

// Allgemeine Server-Parameter

```
begin APDU:apduprops(1);Servereinstellungen
  cov_process_id: R = 5000
  use_confirmed_notification: R = 1
  default_cycletime: R = 15          //seconds
  default_use_cov: R = 0
  default_use_multi_read_write: R = 0
  default_resubtime: R = 1800        //seconds
  resubtimefaktor: W = 1.5
  default_writing_prio: R = 16
  default_polldelay: R = 10
  default_VTI4: R = 1
  default_VTR4: R = 1
  default_BSTR: R = "NULL"
  whois_intervall: W = 60            //seconds
  whois_delay: W = 20                //milliseconds
  alive_increment: W = 5
  alive_decrement: W = 1
  alive_max: W = 20
  dbg_level: W = 0
  dbg_logfile_location: R = "c:\bacopc0.log"
  poll_on_subscribe: R = 0
  serverdeviceID: R = 157751
```

// Device-Parameter

```
begin device:device(2100);device(2100);
  cycletime = 3          //seconds
  useCOV = 0
  //useMultiReadWrite = 0
  //resubtime = 1800       //seconds
  //writePrio = 16
  //polldelay = 20

  description : R = ""
  object-name : R = "MUT_ISP01_RLT_1300"
```

// Objekt-Parameter

```
 begin object:analog-input(5);analog-input(5);cycletime =20
  description : R = "Kanaldruckgeber Zuluft  RLT Anl.05 Druckbelüftung TH"
  object-name : R = "HH_4U2HC_H1_RLT05_00PZU001MA"
  present-value : R = 0.
  status-flags : R = 0100b
```

// Objekt-Parameter

```
begin object:binary-value(1722);binary-value(1722)
```

description : R
object-name : R
present-value : W
status-flags : R

// **Objekt-Parameter**

begin object:analog-value(17);analog-value(17);cycletime =1800
description
object-name
present-value : W
status-flags

These BACnet object types are supported:

analog-input
analog-output
analog-value
binary-input
binary-output
binary-value
multi-state-input
multistate-input
multi-state-output
multistate-output
multi-state-value

3.3 Description of parameters

3.3.1 General Server-parameters

```
begin APDU:apduprops(1);Servereinstellungen
```

„begin APDU“ identifies the section.
“apduprops(1)” identifies this block. As there is only one APDU-block, this name is in fact redundant, but to keep conformity to server and object definitions it has been introduced.

The parameters follow – they will become pseudo properties in the pseudo server “Server” and are accessible via OPC.

cov_process_id (default 5000)

This is a arbitrarily selectable number. BACnet server use this value to distinguish several clients running on the same physical machine.

default_cycletime [Sekunden] (default 15)

The default_cycletime time is used as the Device or Object cycletime when it is not specified in the configuration file for a Device or Object.
The cycletime for a Device or Object is the time interval in which the Server requests a property from the BACnet Device or Object. This is valid also for the Device properties.

default_use_cov (default 0)

The default_able_to_cov value is used as the Device ableToCOV value when it is not specified in the configuration file for a Device.
The ableToCOV parameter specifies the ability of a Device to COV or to poll the data points.
Valid values for the default_able_to_cov and ableToCOV parameters are:
0 = data points must be polled
1 = the device is able to COV and you can use the subscription service

use_confirmed_notification (default 0)

0 = subscribe unconfirmed COV
1 = subscribe confirmed COV

default_use_multi_read_write: (default 0)

The default_use_multi_read_write value is used as the Device default_use_multi_read_write value when it is not specified in the configuration file for a Device.
The default_use_multi_read_write parameter specifies the ability of a Device to support multiple read/writes for its' properties.
Valid values for the default_use_multi_read_write parameters are:

0 = read/write property multiple is not supported
1 = service is supported

Note: This feature is currently not implemented!

default_resubtime [Sekunden] (default 1800)

The default_resubtime time is used as the Device resubtime when it is not specified in the configuration file for a Device.
The resubtime for a Device is the time interval at which the Server will re-subscribe for a COV service from that Device. When a BACnet client (in this case the Amann BACnet/OPC Server) uses a COV service to receive data from a Device it must re-subscribe periodically to the requested COV data points. If the BACnet client fails to re-subscribe the BACnet Device won't report any changes of value (COV) of this object. The resubtime is valid only if the Device ableToCOV parameter is set to 1.

resubtimefaktor (default 1.5)

The Resubtimefaktor parameter defines the number of resubtime intervals the Device will wait before it cancels the Servers subscription. This allows for times when the Server is unable to re-subscribe to the object in time due to unexpected high loads or long operations.

default_writing_prio (default 16)

The default_writing_prio value is used as the Device writePrio value when it is not specified in the configuration file for a Device.
The writePrio parameter specifies the priority level the Amann BACnet/OPC Server will use for write operations to a Device. Valid values for the default_writing_prio and writePrio parameters are:
1 = highest priority
2
.
.
16 = lowest priority

default_polldelay [ms] (default 10)

The default_polldelay time is used as the Device polldelay time when it is not specified in the configuration file for a Device.
The polldelay time specifies the minimum period the Amann BACnet/OPC Server will leave between two poll and/or subscribe events to a Device. It is used to ensure that "slow" Devices will not be swamped by poll and subscribe events

default_VTI4 (default 0)
default_VTR4 (default 0.0)
default_BSTR (default "not inizialized")

The **default_VTI4, default_VTR4** and **default_BSTR** parameters are all very similar in function. These parameters define the start up values (i.e. value before update from a Device) of each of the different OPC data types used by the Amann BACnet/OPC Server. As stated earlier, as the Server starts information about BACnet Devices and objects is read from the "cnf.srv" configuration file. As the namespace is created values of properties like status-flag, present-value and object-name are normally initialised to 0, 0.0 or "0" depending on their type. This default initial value of 0, 0.0 or "0" can be changed by setting **default_VTI4, default_VTR4** and **default_BSTR** to another value. The following example may clarify the usefulness of these parameters.

Perhaps alarms in your project are "ON" as the OK state. This could potentially create a problem at start up of the Server, as all alarm values would initially be set to "OFF" until updated from the BACnet device. By setting **default_VTI4** to "1" then all Binary Points (and Multistates) will be initialised to "ON" (or "1") thus solving this problem.

Parameter	General type	OPC type	BACnet example
default_VTI4	Integer, Boolean	VT_I4	binary-value.present-value
default_VTR4	Float	VT_R4	analog-value.present-value
default_BSTR	String	VT_BSTR	analog-value.object-name

whois_intervall [Sekunden] (default 60)

The Server creates an Online_Status OPC data point for each Device. The Online_Status data point value indicates whether communications to the Device is online (Online_Status = 1) or Offline (Online_Status = 0). The Server uses a BACnet service called "Who Is" to periodically check communications with a BACnet Device. The interval between "Who Is" requests to the same BACnet Device is the whois_intervall time in seconds

whois_delay [ms] Sekunden (default 20)

The **whois_delay** is the minimum time period between any two “Who Is” requests, the two “Who Is” requests are not likely to be to the same BACnet Devices. The **whois_delay** enables optimisation or tuning out of the BACnet traffic bottlenecks. Note the **whois_delay** time is in milliseconds.

alive_increment (default 5), alive_decrement (default 1), alive_max (default 15)

The Server has an online counter for each BACnet device. For each “Who Is” request to a BACnet Device its online counter is decreased by the value of the alive_decrement parameter. For each “I Am” reply from a BACnet Device its online counter is increased by value of the alive_increment parameter. The alive_max parameter sets the maximum value that an online counter can be increased to. The Online_Status of a BACnet Device is set to 0 (Offline) when its online counter = 0 and set to 1 (Online) if its online counter > 0.
The Server initialises all online counters to 0 at start up. If the Online_Status of a BACnet Device is 0 (Offline) then no requests (other than “Who Is” requests), polls or writes are sent from the Server to the BACnet Device. Logically when the BACnet Device is Offline it will not answer; thus BACnet traffic will be reduced.

dbg_level (default 0)

The dbg_level parameter determines what level of debugging information will be written into the debug log files.

0	no debug information
255	all debug information

There are several kinds of debug messages and they are separated into the following 4 categories:

ERROR	128
GENERAL	64
OPC	32
BAC	16

These values are designed to get bit or’d.

dbg_logfile_location (default „c:\\bacopc.log“)

The dbg_logfile_location parameter determines the location where the debug log files will be found. A double-backslash “\\” must be used to separate directories. The 10 (max) rotating debug log files could use as much as 200 Mbytes of space.

serverdeviceID (default: calculated from nic address)

As defined in the BACnet standard all Devices on a BACnet network must have a unique BACnet ID to identify themselves. The ServerdeviceID parameter sets the BACnet ID for the Server.

poll_on_subscripe (default 0)

Normally a Device will automatically send a “Notification” after the subscription to the COV service of an Object value by the Server, unfortunately some devices don’t do this. The poll_on_subscripe parameter determines if Objects will be polled once after subscription to ensure the initialised data point in the Server is updated.

0	no poll on subscription
1	poll once on subscription

3.3.2 Device-Parameter

```
begin device:device(2100);device(2100)
```

„begin device“ identifies the section.

The first “device(2100)” declare a BACnet device reference to the device 2100.

The second device(2100) defines the OPC reference of the BACnet device 2100 (you can change this to any identifier you like, bacbrowse will generate this name).

cycletime
useCOV
useMultiReadWrite
resubtime
writePrio
polldelay

These parameters correspond to the default_xxxx global parameters and will override these if specified.

3.3.3 Object-Parameter

```
   begin object:analog-input(5);analog-input(5);cycletime =20
```

„begin object” identifies the section.

The first “analog-input(5)” defines the object identifier of the BACnet object.

The second “analog-input(5)” is the OPC reference name analogous to the device definition.

cycletime

This is the only attribute which can get redefined on object level. Thus it is possible to poll a temperature once per 10 minutes, but some other time critical value once a second (presumed the total load of the network isn’t too high).

Following the effective BACnet properties:

Analog Input	Analog Output	Analog Value
cov_increment	cov_increment	cov_increment
object_identifier	object_identifier	object_identifier
object_type	object_type	object_type
object_name	object_name	object_name
description	description	description
present_value	present_value	present_value
reliability	reliability	reliability
status_flags	status_flags	status_flags
event_state	event_state	event_state
out_of_service	out_of_service	out_of_service
units	units	units
time_delay	priority_array	priority_array
high_limit	relinquish_default	relinquish_default
low_limit	time_delay	time_delay
deadband	high_limit	high_limit
limit_enable	low_limit	low_limit
event_enable	deadband	deadband
notification_class	limit_enable	limit_enable
acked_transitions	event_enable	event_enable
notify_type	notification_class	notification_class
event_time_stamps	acked_transitions	acked_transitions
	notify_type	notify_type
	event_time_stamps	event_time_stamps

Binary Input	Binary Output	Binary Value
object_identifier	object_identifier	object_identifier
object_type	object_type	object_type
object_name	object_name	object_name
active_text	active_text	active_text
inactive_text	inactive_text	inactive_text
description	description	description
present_value	present_value	present_value
reliability	reliability	reliability
status_flags	status_flags	status_flags
event_state	event_state	event_state
out_of_service	out_of_service	out_of_service
polarity	polarity	polarity
time_delay	priority_array	priority_array
alarm_value	relinquish_default	relinquish_default
event_enable	time_delay	time_delay
notification_class	feedback_value	alarm_value
acked_transitions	event_enable	event_enable
notify_type	notification_class	notification_class
event_time_stamps	acked_transitions	acked_transitions
	notify_type	notify_type
	event_time_stamps	event_time_stamps

Multistate Input	Multistate Output	Multistate Value
object_identifier	object_identifier	object_identifier
object_type	object_type	object_type
object_name	object_name	object_name
number_of_states	number_of_states	number_of_states
state_text	state_text	state_text
description	description	description
present_value	present_value	present_value
reliability	reliability	reliability
status_flags	status_flags	status_flags
event_state	event_state	event_state
out_of_service	out_of_service	out_of_service
time_delay	priority_array	priority_array
alarm_values	relinquish_default	relinquish_default
fault_values	time_delay	time_delay
event_enable	feedback_value	alarm_values
notification_class	event_enable	fault_values
acked_transitions	notification_class	event_enable
notify_type	acked_transitions	notification_class
event_time_stamps	notify_type	acked_transitions
	event_time_stamps	notify_type
		event_time_stamps

Device
number_of_apdu_retries
apdu_timeout
daylight_savings_status
utc_offset
segmentation_supported
max_apdu_length_accepted
protocol_object_types_supported
protocol_conformance_class
protocol_version
description
location
application_software_version
firmware_revision
model_name
vendor_identifier
protocol_services_supported
object_name
object_identifier
object_type
system_status
vendor_name
local_date
local_time
protocol_revision
device_address_binding

4 Server start up and shutdown

If required by an OPC client the Amann BACnet/OPC Server will start automatically and will run as long as a Client is connected. It is also possible to start the Amann BACnet/OPC Server manually and connect the OPC Clients later. The Server can be stop only if all connected OPC Clients are disconnected. The running Server is indicated by a symbol (see figure 4.1) on the Task bar.

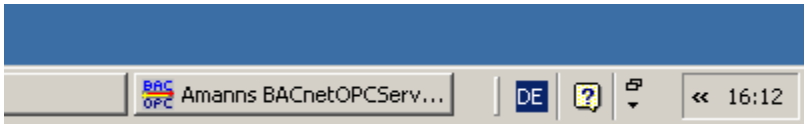


Figure 4.1. Indication that the Server is running.

The Server status information is shown in a panel at the bottom of the maximised window (see figure 4.2). The following status parameters can be seen:

Server running since	Server start up date and time.
Devicecount	Number of BACnet devices (including the Server) that the Server has been configured to communicate with.
Objectcount	Number of BACnet Objects that the Server has been configured to communicate with.
Propertycount	Number of Properties (including the Server Properties) that the Server has been configured to communicate with.
Object Max	The maximum number of BACnet Objects that can be accessed at the same time by the Server (license dependent).
active OPC Items	Number of OPC Objects that are being requested by all OPC Clients.
active BACnet Objects	Number of BACnet Objects requested by all OPC Clients. This value can be less than that of the active OPC Items because more then one OPC Client can be accessing the same BACnet Object at the same time.

Figure 4.2. Server status information shown in bottom panel of the maximised window

5 Runtime

5.1 Log files

At runtime activities (dependent on the debug level) of the Server are logged. With the aid of the application **DBWIN32.exe** the debug information can be issued live. In addition to this rotating log files are written in the directory configured in the parameter **dbg_logfile_location**. A maximum of 10 files each with a maximum size of 20 MB will be written. The entries in the log file are divided into four classes:

Class	Label in log file	Description
Error	ERROR	Global error
General	GENERAL	General announcements about the Server
BACnet	BACnet	Announcements of BACnet specific traffic
OPC	OPC	OPC specific announcements

A table of debug messages follows:

Meldungen Logdatei Amann BACnet/OPC-Server	Error	General	BACnet	OPC
Trying to start ioThreadMain		X		
BACless Version, no bacnet threads get started		X		
Trying to start m_queueReadMain		X		
Trying to start m_queueWriteMain		X		
Activating Dev: [dev] Obj: [obj]				X
Deactivating Dev: [dev] Obj: [obj]				X
Unactivated Object deactivated!. inst: [obj]		X		X
Unexpected aq method encountered. Call service!	X		X	
!!!!should never happen: else part in BACDevice::ActiveList::aquire()	X		X	
Beginning iothread		X		
BACstack couldn't be initialized properly.	X		X	
ERROR: BACstack couldn't be started properly.	X		X	
BACnet traffic is enabled			X	
Wri Err from BACnet	X		X	
Wri Ack from BACnet			X	
Writing to BACnet Dev: [dev] Obj: [obj] Prop: [prop] Value: [val] Prio: [prio]			X	
Notific from BACnet Dev: [dev] Obj: [obj9 Prop: [prop] Pid: [pid]			X	
Sub Err from BACnet Dev: [dev] Obj: [obj] Error class: [eclass] Error code: [ecode]	X		X	
Sub Ack from BACnet Dev: [dev] Obj: [obj]			X	
Error from BACnet Dev: [dev] Obj: [obj] Error class: [eclass] Error code: [ecode	X		X	
I am from BACnet Dev: [dev]			X	
Whois to BACnet Dev: [dev] Assumed onlinestatus:[onlst] Score: [score]			X	
Request to BACnet Dev: [dev] Obj: [obj] Prop: [prop]			X	
Subscr to BACnet Dev: [dev] Obj: [obj] Prop: [prop]			X	
Unsubsc to BACnet Dev: [dev] Obj: [obj] Prop: [prop]			X	
Answer from BACnet Dev: [dev] Obj: [obj] Prop: [prop] Value: [val]			X	
Cannot map bac type [bac.type] to opc type [opc.vt]	X	X		
Bitstring too long, cannot convert	X	X		
Cannot map bac type bitstring to opc type VT_BSTR only!	X	X		
Currently maping charstring to VT_BSTR only!	X	X		
Cannot map ObjectIdentifier to opc type	X	X		
BACtype [bac.type] OPCType [opc.vt] mapping failed	X	X		

BACnet-specific Errors are listed in the appendix 6.3 described.

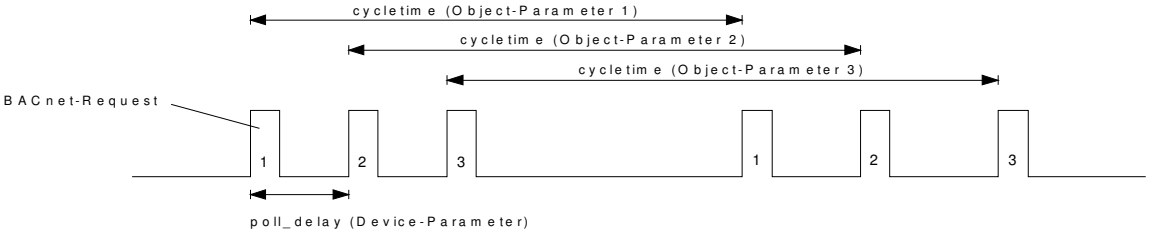
5.2 Online status

Data points from a BACnet Device will only be polled or subscribed if the “Online status” of the device is ONLINE. This has been previously described in the Server parameter **whois_intervall** section.

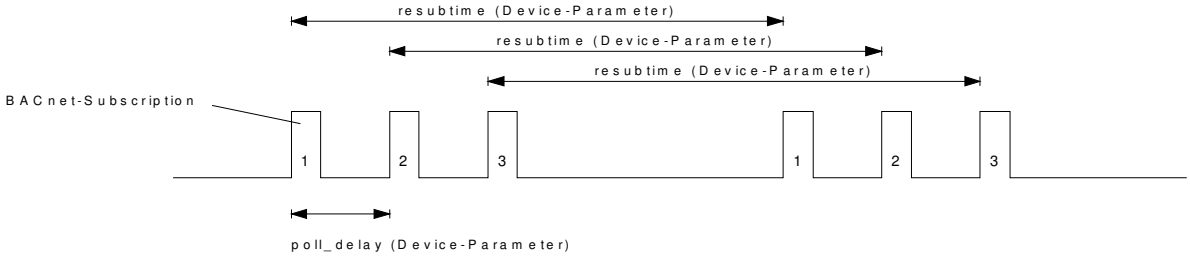
5.3 Polls and COV

In general BACnet-Devices should be operated in “COV” Mode to reduce the BACnet traffic. Traffic is reduced because only changes of Value will cause BACnet traffic. Unfortunately not all BACnet Devices support COV and these Device then have to be polled at regular intervals thus increasing traffic.

Poll Mode



COV Mode



5.4 Demo mode

The demo version of the BACnet/OPC server is restricted to 500 BACnet objects and will shutdown automatically after 48 hours.

6 Appendix

6.1 Default values

0 is logically FALSE
1 is logically TRUE

Parameter	Default-Wert
cov_process_id	5000
use_confirmed_notification	0
default_cycletime	15
default_use_cov	0
default_useMultiReadWrite	0
default_resubtime	1800
resubtimefaktor	1.5
default_writing_prio	16
default_polldelay	10
default_VTI4	1
default_VTR4	1
default_BSTR	"NULL"
whois_intervall	60
whois_delay	20
alive_increment	5
alive_decrement	1
alive_max	20
dbg_level	0
dbg_logfile_location	"c:\bacopc.log"
serverdeviceID	Calculated from nic address
poll_on_subscripe	0

6.2 Files

Following files are installed on your Hard Disk during the installation process:

Default Zielverzeichnis	Name	Version	Aufgabe
c:\Programme\Amann\BACOPC\	BACOPC.exe	1.0.1.0	BACnet/OPC-Server
	SOWatch.exe	3.10.0.300	OPC Watch
	EditConfig.exe	1.0.0.1	Konfigurationsdatei bearbeiten
	DBWIN32.exe	1.3.0.0	Ausgabe von Debugmeldungen
	SOClient.exe	3.10.0.300	OPC Client
	cnf.srv		Beispiel Konfigurationsdatei
c:\programme\winpcap	install.log		Logdatei Netzwerkinstallation
	uninstall.exe		Deinstallationsprogramm Netzwerk
c:\winnt\	Cbndll.dll	2.2.0.1	CRYPTO-BOX Windows 32 bit DLL
	MFC42D.DLL	6.0.8665.0	MFCDLL Shared Library – Debug Version
	MSVCP60D.DLL	6.0.8168.0	MS C++ Runtime Library
	MSVCRTD.DLL	6.0.8797.0	MS C Runtime Library
	alrsvc.dll	4.0.1374.1	Warndienst-DLL
	OpcEnum.exe	1.0.0.4	List installed OPC Servers
	amldb.als		Lizenz-Datei
	tail.exe		Debug-Tool Dateiausgabe
	grep.exe		Debug-Tool Dateien durchsuchen
	MARXDEV1.REG		CRYPTO-BOX Registry-Einträge
	MARXDEV2.REG		CRYPTO-BOX Registry-Einträge
	MARXDEV3.REG		CRYPTO-BOX Registry-Einträge
c:\winnt\system32*	OPCPROXY.DLL	2.0.0.1	OPC Data Access 2.0 Proxy
	alcsrv.dll	1.0.0.0	Amann Lizienz-Server
	opc_aeps.dll	1.0.5.0	OPC Alarm and Event Server Proxy/Stub
	opccomn_ps.dll	1.0.0.2	OPC Common Interfaces 1.0 Proxy
	SOProxy.dll	3.5.0.263	OPC Interfaces Proxy
	ACTXPRXY.DLL**	4.70.12.15	ActiveX Proxy
	packet.dll	2.3.0.33	WinPcap low level packet library
	wpcap.dll	0.6.2.0	WinPcap high level library (nicht benutzt)
C:\WINDOWS\System32\drivers\	npf.sys	2.3.0.34	Netgroup Packet Filter Driver
	MARXDEV1.SYS	2.4.0.1	CRYPTO-BOX 560/VERSA driver for NT
	MARXDEV2.SYS	2.4.0.1	CRYPTO-BOX 560/VERSA driver for NT
	MARXDEV3.SYS	2.4.0.1	CRYPTO-BOX 560/VERSA driver for NT

6.3 BACnet-specific Error-codes

Code	Bedeutung
0	other
1	authentication_failed
2	configurationinprogress
3	devicebusy
4	dynamic_creation_not_supported
5	file_access_denied
6	incompatible_security_levels
7	inconsistentparameters
8	inconsistentselectioncriterion
9	invaliddatatype
10	invalid_file_access_method
11	invalid_file_start_position
12	invalid_operator_name
13	invalidparameterdatatype
14	invalid_time_stamp
15	key_generation_error
16	missingrequiredparameter
17	noobjectsofspecifiedtype
18	nospaceforobject
19	no_space_to_add_list_element
20	no_space_to_write_property
21	novtsessionsavailable
22	property_is_not_a_list
23	object_deletion_not_permitted
24	object_identifier_already_exists
25	operationalproblem
26	password_failure
27	readaccessdenied
28	security_not_supported
29	servicerequestdenied
30	timeout
31	unknownobject
32	unknownproperty
33	
34	unknownvtclass
35	unknownvtsession
36	unsupportedobjecttype
37	value_out_of_range
38	vtsessionalreadyclosed
39	vtsessionterminationfailure
40	writeaccessdenied
41	character_set_not_supported
42	invalid_array_index

7 Error handling

Symptom	Possible cause	Fix
The BACnet/OPC Server won't shut down	OPC Clients are still connected.	Look in the Task-Manager for running OPC Clients and shut them down
The BACnet/OPC Server won't start	The Registration has failed	Please run the Setup again.
No Logfiles are written	Debuglevel = 0	Please change the Debuglevel